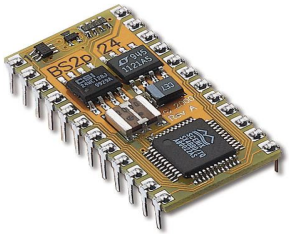


Capítulo No. 07
Pantalla de Caracteres LCD

PANTALLA LCD

Introducción. En capítulos anteriores, se han utilizado como dispositivos de visualización los diodos LED y la pantalla de LEDs de siete segmentos. Estos han demostrado ser sencillos, eficientes y económicos como elementos de visualización. Sin embargo, existirán ocasiones donde se necesite visualizar una gran cantidad de información o texto alfanumérico, y es precisamente aquí, donde la pantalla de cristal líquido (LCD) lo puede hacer mejor que cualquier otro dispositivo. En este capítulo, se hará una explicación detallada acerca del funcionamiento de una pantalla LCD de caracteres, la misma, se utilizará de ahora en adelante en casi la totalidad de los experimentos.

La Pantalla LCD es uno de los periféricos más empleados para la presentación de mensajes, variables y casi cualquier información proveniente de un microcontrolador. Gracias a su flexibilidad, buena visibilidad y precio reducido se ha introducido de manera determinante en el mercado.

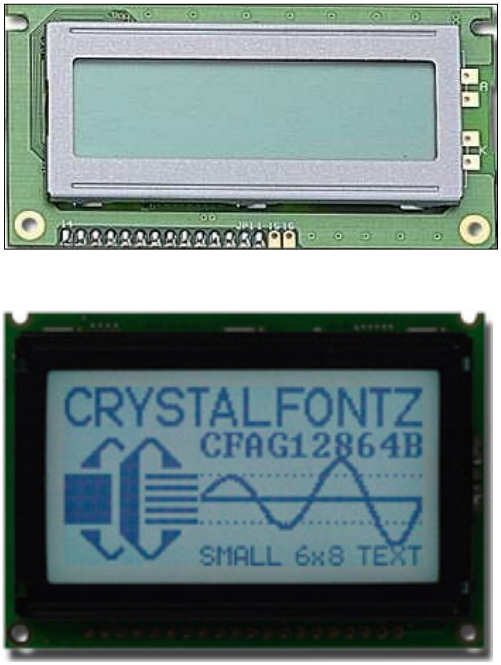


Figura No. 7.1: Arriba. pantalla LCD Alfanumérica de 2 x 16. Abajo, pantalla LCD Gráfica de 128 x 64 píxel.
Fuente: http://www.crystalfontz.com.

Vocabulario sobre LCD

LCD Alfanumérica
Pantalla en la cual, se pueden presentar caracteres y símbolos especiales en las líneas predefinidas del LCD. Su especificación viene dada como cantidad de caracteres por columna y número de filas del display. Por ejemplo: 2 x 16, 4 x 20.

LCD Gráfica
Pantalla en la cual, se pueden presentar caracteres, símbolos especiales y gráficos. Su especificación viene dada en píxeles. Por ejemplo 128 x 64.

LCD Paralela
Los datos y comandos son enviados a través de un bus de datos paralelo, ya sea en modo de 4 o de 8 bits.

LCD Serial
La pantalla es capaz de recibir la información de manera serial asincrónica utilizando sólo 2 líneas para este fin. Toda pantalla paralela puede ser convertida a serial mediante un circuito especializado, como por ejemplo el Controlador serial para LCD modelo SLCD-IC.

Las pantallas de LCD poseen un controlador empotrado dentro de su propio módulo, el cual, se encarga de gestionar el control de sus terminales para la presentación de los caracteres y almacena el conjunto predefinido de caracteres en una memoria no volátil. El controlador más utilizado para las pantallas alfanuméricas es el HITACHI 44780.

Las pantallas LCD que incluyen un controlador interno y dedicado como el HITACHI 44780 pueden:

- Presentar caracteres ASCII y griegos.
- Recibir y mostrar caracteres personalizados.
- Desplazar los caracteres (*scroll*) tanto a la derecha como a la izquierda.
- Cambiar el aspecto y el movimiento del cursor.
- Direccionar de manera simple la posición de visualización de los caracteres.

Los terminales de conexión de las pantallas LCD de caracteres han sido estandarizados, siendo generalmente compatibles pin a pin con la tabla No. 7.1.

Terminal	Nombre	Función	Descripción
1	Vss	Energía	Referencia 0 V. GND
2	Vdd	Energía	+5 V DC
3	Vee	Ajuste Contraste	Variable de 0 a 5 V
4	RS	Comando	Selección de Dato/Comando
5	R/W	Comando	Control de Lectura/Escritura
6	E	Comando	Habilitación
7	D0	E/S	DATO LSB
8	D1	E/S	DATO
9	D2	E/S	DATO
10	D3	E/S	DATO
11	D4	E/S	DATO
12	D5	E/S	DATO
13	D6	E/S	DATO
14	D7	E/S	DATO MSB

Tabla No. 7.1: Terminales de Conexión para una LCD alfanumérica estándar.
Fuente: The Extended Concise LCD Data Sheet for HD44780, http://www.beyondlogic.com.

Vocabulario sobre LCD

CGRAM
(Character Generator RAM)

Memoria volátil de 64 bytes que permite almacenar hasta 8 caracteres personalizados para ser mostrados en la pantalla.

DDRAM (Display Data RAM)

Memoria volátil de 80 bytes, en la cual, se almacenan los caracteres que se van a mostrar en la pantalla.

CGROM
(Character Generator ROM)

Memoria no volátil que almacena los caracteres predefinidos soportados por la pantalla y su controlador.

Ocho Bits
Modo de trabajo en el que se utilizan todos los bits para la transmisión de los datos y los comandos hacia una LCD.

Cuatro Bits
Modo de trabajo en el cual la transferencia de información se efectúa por medio de los cuatro (4) bits más significativos.

El modo de trabajo de 4 bits permite reducir el número de terminales del microcontrolador destinados a la interfaz del LCD, pero se obtiene una menor velocidad de transferencia de la información.

La información que se envía a la pantalla LCD puede interpretarse de dos maneras: como un **comando,** cuando se coloca al terminal denominado RS en nivel bajo. Y como un **carácter,** cuando el terminal RS está en nivel alto. Un comando es un dato numérico de 8 bits que le indica al controlador HITACHI 44780 o compatible, que debe realizar una cierta operación.

Comandos	Valor Numérico	Descripción
Nulo	0	No se efectúa ninguna operación especial.
Limpiar LCD	1	Limpia la pantalla y desplaza el cursor a la posición 0 (*Home*).
Home LCD	2	Mueve el cursor a la posición de inicio.
Incrementar cursor	6	Mueve el cursor a la derecha sin desplazar la pantalla.
Off LCD	8	Apaga al LCD sin perder los datos que esta tenía.
On LCD	12	Enciende al LCD sin mostrar el cursor y con el último contenido almacenado en buffer.
Parpadeo del cursor	13	Enciende el LCD con el cursor parpadeando.
Cursor subrayado	14	Enciende el LCD con el cursor con un subrayado.
Cursor izquierda	16	Mueve el cursor un carácter a la izquierda.
Cursor derecha	20	Mueve el cursor un carácter a la derecha.
Scroll izquierda	24	Desplaza la pantalla 1 carácter a la izquierda.
Scroll derecha	28	Desplaza la pantalla 1 carácter a la derecha.
Mover a dirección CGRAM	64 + dirección	Mueve el apuntador a una localidad de la memoria de generación de caracteres (Character Generator RAM).
Mover a dirección DDRAM	128 + dirección	Mueve el cursor a una localidad de la memoria de visualización de caracteres (Display Data RAM).

Tabla No. 7.2: Comandos de uso común de un LCD.
Fuente: PBASIC *Syntax Guide*, Parallax 2003.

	000	016	032	048	064	080	096	112	128	144	160	176	192	208	224	240
00	*1			0	@	P	`	p				—	タ	ミ	α	ρ
01	*2		!	1	A	Q	a	q			｡	ｱ	チ	ム	ä	q
02	*3		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
03	*4		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
04	*5		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
05	*6		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
06	*7		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
07	*8		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
08	*1		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x
09	*2		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
10	*3		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
11	*4		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
12	*5		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
13	*6		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
14	*7		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
15	*8		/	?	O	_	o	←			ッ	ソ	マ	°	ö	■

Tabla No. 7.3: *Set* de caracteres para un Hitachi 44780.
Fuente: Manual del controlador HD44780.

Aún cuando los comandos de la tabla No. 7.2 son suficientes para manejar la pantalla LCD en la mayoría de las aplicaciones, el programador puede crear sus propios comandos de control, utilizando para ello las especificaciones que suministra el fabricante. En el caso de las pantallas que utilizan el controlador HITACHI 44780, se puede emplear la tabla No. 7.5.

	Posiciones Visibles																Posiciones no Visibles		
Línea 1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	39
Línea 2	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	...	103

	Posiciones Visibles																			
Línea 1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Línea 2	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
Línea 3	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Línea 4	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103

Tabla No. 7.4: Posiciones en una pantalla LCD de 2x16 y 4x20. Números en Hexadecimal.
Fuente: PBASIC *Syntax Guide*, Parallax 2003.

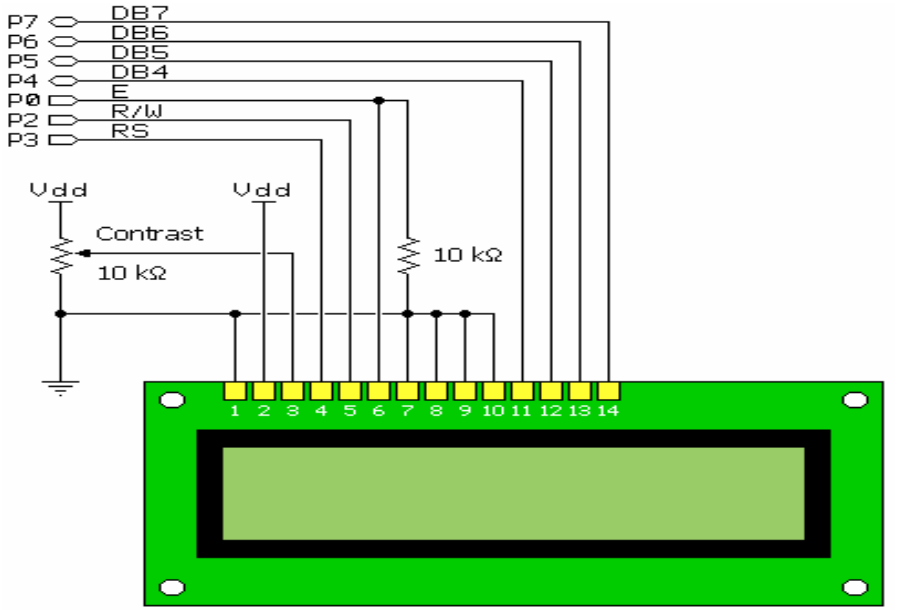


Figura No. 7.2: Esquema de conexiones entre una pantalla LCD y un BASIC Stamp modelo BS2p.
Fuente: PBASIC *Syntax Guide*, Parallax 2003.

Tabla No. 7.5: Registro de configuración para el controlador HITACHI 44780.
Fuente: *The Extended Concise LCD Data Sheet for HD44780*, http://www.beyondlogic.com.

Instrucción	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Descripción
Modo de entrada	0	0	0	0	0	0	0	1	I/D	S	Dirección del cursor (I/D). Auto desplazamiento de la pantalla(S)
Control de pantalla y del cursor	0	0	0	0	0	0	1	D	C	B	Encendido/Apagado de la LCD (D) y del cursor (C). Parpadeo del cursor (B).
Desplazamiento	0	0	0	0	0	1	S/C	R/L	x	x	Desplaza la pantalla o el cursor (S/C) en la dirección especificada por (R/L).
Selección de Funciones	0	0	0	0	1	DL	N	F	x	x	Ancho del bus de datos (DL). Número de líneas (N). Tamaño de la letra (F).
Mover a dirección CGRAM	0	0	0	1	Dirección						Mueve el apuntador de la CGRAM a la posición indicada.
Mover a dirección DDRAM	0	0	1	Dirección							Mueve el apuntador de la DDRAM a la posición indicada.
Escribir al LCD	1	0	Dato								Escribe información (CGRAM o DDRAM).
Leer desde el LCD	1	1	Dato								Lee información (CGRAM o DDRAM).

Descripción de los Bits		
I/D	1 0	Movimiento del cursor a la derecha Movimiento del cursor a la izquierda
S	1 0	Desplazamiento automático de la pantalla: SI Desplazamiento automático de la pantalla: NO
D	1 0	Pantalla encendida Pantalla apagada
C	1 0	Cursor visible Cursor invisible
B	1 0	Cursor parpadeando Cursor sin parpadeo
S/C	1 0	Desplazamiento de la pantalla Desplazamiento del cursor
R/L	1 0	Desplazamiento a la derecha Desplazamiento a la izquierda
DL	1 0	Modo de 8 bits Modo de 4 bits
N	1 0	2 líneas 1 línea
F	1 0	5x10 5x7

MÁS INSTRUCCIONES DE PBASIC

LCDCMD

Envía un comando a una pantalla LCD.

LCDCMD *Pin, Comando*

- ***Pin*** **:** Es una variable, constante o expresión (0 - 1 ó 8 - 9) que indica cuales de los pines de E/S serán usados para manipular la pantalla LCD.
- ***Comando*****:** Es una variable, constante o expresión (0 - 255) con el valor del comando a enviar al LCD, ver tabla No. 7.2.

LCDIN

Recibe información proveniente de una pantalla LCD y es empleada para leer el contenido de las memorias DDRAM y CGRAM.

LCDIN *Pin, Dirección, [Datos]*

- ***Pin*****:** Es una variable, constante o expresión (0 - 1 ó 8 - 9) que indica cuales de los pines de E/S serán usados para manipular la pantalla LCD.
- ***Dirección*****:** Es una variable, constante o expresión (0 - 255) que indica la dirección de memoria RAM de la pantalla LCD que se quiere leer.
- ***Datos*****:** Es un listado de variables con sus respectivos modificadores de formato que definen como será interpretada la información entrante.

LCDOUT

Envía un comando, seguido de al menos un byte de datos al LCD.

LCDOUT *Pin, Comando, [Datos]*

- ***Pin*****:** Es una variable, constante o expresión (0 - 1 ó 8 - 9) que indica cuales de los pines de E/S serán usados para manipular la pantalla.
- ***Comando*****:** Es una variable, constante, expresión (0 - 255) con el valor del comando a enviar al LCD.
- ***Datos*****:** Es una lista de variables, constantes o expresiones con sus respectivos modificadores que indican la forma en que serán transmitidos a la pantalla.

LOOKUP

Efectúa la búsqueda de un elemento, dado su índice, dentro de una lista.

LOOKUP *Indice, [Valor0, Valor1, ... ValorN] , Variable*

- ***Indice*:** Es una variable, constante o expresión en el rango de 0 a 255 que indica la posición dentro de la lista del elemento que se desea extraer.
- ***Valor0, Valor1 ...ValorN* :** Son variables, constantes o expresiones de 0 a 65535, que componen la lista.
- ***Variable:*** Es una variable en el rango de 0 a 65535 que contendrá el valor del elemento encontrado en la posición ***Indice.***

Si ***Indice*** excede el tamaño de la lista de elementos, ***Variable*** no cambiará su valor.

LOOKDOWN

Permite hacer la búsqueda de un elemento dentro de una lista de acuerdo a un criterio y retorna la ubicación del primer elemento que satisface la condición.

LOOKDOWN *Elemento*, {Operador Condicional} *[Valor0, Valor1, ... ValorN], Variable*

- ***Elemento*:** Variable, constante o expresión (0 - 65535) que constituye el elemento a comparar.
- **Operador Condicional:** Operador relacional que puede ser usado como criterio durante la comparación. Si se omite se utilizará el igual (=) que es el predeterminado.
- ***Valor0, Valor1...ValorN*** : Son variables, constantes o expresiones de 0 a 65535, que componen la lista contra la que se compara a ***Elemento***.
- ***Variable***: Es una variable en el rango de 0 a 255, que contendrá el valor del índice del primer elemento que satisface la comparación. Si la comparación resultara falsa para todos los elementos de la lista; ***Variable*** no cambiará su valor.

Los operandos que intervengan en la instrucción **LOOKDOWN** deben ser números enteros sin signo.

DATA

Permite escribir información en la EEPROM durante el proceso de descarga del programa al BASIC Stamp.

{Simbolo} DATA {@Dirección,} {Word} ***Dato {, Dato ...}***

- ***Símbolo*****:** Es un parámetro opcional. Consiste en un nombre simbólico que será definido automáticamente como una constante igual a la localidad de memoria donde se almacene el primer dato.
- ***Dirección*****:** Es un parámetro que indica la localidad de inicio donde se desea grabar los datos.
- ***Dato*****:** Constante o expresión de 0 a 65535, que se desea almacenar en la EEPROM del BS2p.
- **Word:** Modificador opcional para manipular variables de 16 bits.

READ

Lee una o más localidades de la memoria EEPROM y almacena en una o más variables el contenido de la(s) misma(s).

READ ***Localidad,*** **{Word}** ***Variable*** **{,** **{Word}** ***Variable ...}***

- ***Localidad*****:** Es una variable, constante o expresión de 0 a 2047 que indica la dirección de la EEPROM cuyo contenido se quiere leer.
- ***Variable*****:** Es una variable (generalmente tipo byte) donde se almacena el contenido de la localidad leída.
- **Word:** Modificador opcional para manipular variables de 16 bits.

EJERCICIO # 1: CONOCIENDO E INICIANDO AL LCD

Objetivo: Manipular una pantalla de cristal líquido paralela de 2x16 como elemento de visualización.

Elementos a Utilizar

(1) BASIC Stamp 2p.
(1) LCD alfanumérico 2 x 16 tipo paralelo.
(1) Resistencia de 10 KOhms.
(1) Potenciómetro de 10 Kohms.
(1) Banco de Pulsadores.
(1) Un cable serial DB9.
(1) StampW (v2 Beta 2.1 o superior).

Monte en el protoboard que acompaña al sistema de desarrollo MDK-B28, el esquema de la figura No. 7.3.

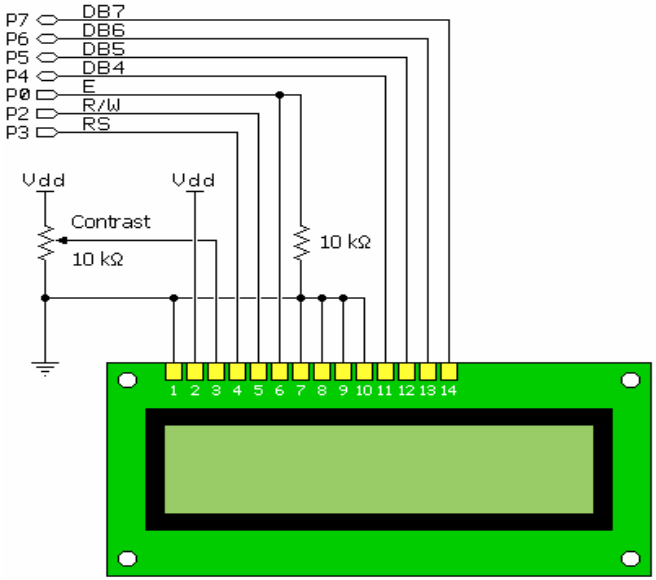


Figura No. 7.3: Conexión del LCD al microcontrolador para el ejercicio # 1.
Fuente: BASIC Stamp® PBASIC Online Help.

Antes de energizar el sistema de desarrollo, verifique **cuidadosamente** las conexiones pues podría **destruir** la pantalla. Los circuitos de la figuras No. 7.3 y No. 7.4 serán utilizados durante los 4 ejercicios de este actividad, por favor **no los desmonte**. Ajuste el potenciómetro en un nivel medio e introduzca al microcontrolador el programa 7_1.bsp.

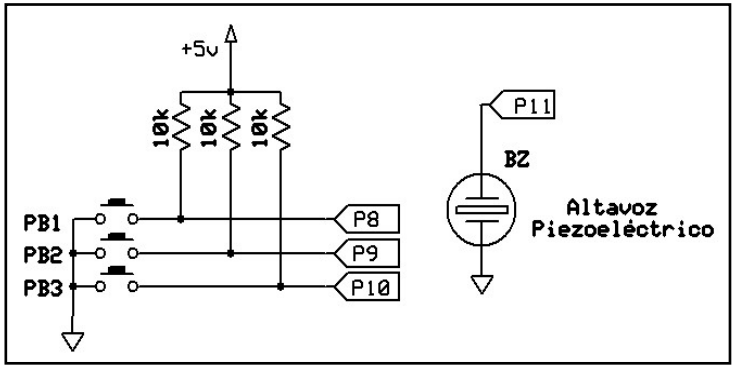


Figura No. 7.4: Conexión de pulsadores y altavoz para el ejercicio # 1.
Fuente: Elaboración Propia.

```
'Programa 7_1.bsp
'Este programa tiene por objetivo mostrar:
'1) El uso de la pantalla LCD como dispositivo de visualización
'2) El uso de LCDCMD y LCDOUT para enviar comandos y mensajes a la pantalla LCD
'3) Ejecutar la rutina de inicialización de la pantalla LCD
'================================================================================
'{$STAMP BS2p}
'{$PBASIC 2.5}
'--------------------------------------------------------------------------------
'Declaración de Variables y Constantes

BorrarLCD     CON %00000001       'Limpiar pantalla
MoverCrsr     CON %10000000       'Posicionar el cursor
Despertar     CON %00110000       'Despertar al LCD
Modo4bits     CON %00100000       'modo de 4 bits de datos
Modo2linea    CON %00101000       'modo de 2 líneas, tamaño de 5x8
LCDOFF        CON %00001000       'Apagar la pantalla
LCDON         CON %00001100       'Encender, con cursor la pantalla
IncrCrsr      CON %00000110       'Auto-incremento del cursor

Pulset1       VAR Byte            'Espacio de trabajo de PB1
Pulset2       VAR Byte            'Espacio de trabajo de PB2
Pulset3       VAR Byte            'Espacio de trabajo de PB3
Pos           VAR Byte            'Variable de uso general
x             VAR Nib             'Variable de uso general

LCDPin        PIN 0
PB1           PIN 8               'Pulsador PB1
PB2           PIN 9               'Pulsador PB2
PB3           PIN 10              'Pulsador PB3
BZ            PIN 11              'Salida al piezoeléctrico
LED           PIN 12              'Salida al LED

'--------------------------------------------------------------------------------
'Programa Principal:
Inicio:
  GOSUB IniLcd                    'Inicialización de la pantalla
  LOW LED                         'Apagar el LED conectado a P12
```

Continuación del Programa 7_1.bsp

```
Menu:
  LCDCMD LCDPin ,BorrarLCD      'Limpiar la pantalla
  DO
    LCDOUT LCDPin ,MoverCrsr + 3,["***MENU****"]      'Título del Menú
    LCDOUT LCDPin ,MoverCrsr+64,["1 BZ 2 LED 3 MSG"]  'Items del Menú
    BUTTON PB1,0,255,0,pulset1,1,PS1               'Si PB1 fue presionado va a PS1
    BUTTON PB2,0,255,0,pulset2,1,PS2               'Si PB2 fue presionado va a PS2
    BUTTON PB3,0,255,0,pulset3,1,PS3               'Si PB3 fue presionado va a PS3
  LOOP
  PS1:
    LCDCMD LCDPin ,BorrarLCD                       'Limpiar pantalla
    LCDOUT LCDPin ,MoverCrsr+64,["Sonido..."]      'Enviar mensaje a la posición
    FREQOUT BZ,1500,1200                           'Generar el sonido
    PAUSE 1000
    GOTO Menu
  PS2:
    LCDCMD LCDPin ,BorrarLCD                       'Limpiar la pantalla
    LCDOUT LCDPin ,MoverCrsr+2,["OBSERVE COMO"]    'Mostrar mensaje
    LCDOUT LCDPin ,MoverCrsr+64+3,["P12 CAMBIA"]
    TOGGLE LED                                     'Cambiar el estado del LED
    PAUSE 1000
    GOTO Menu
  PS3:
    FOR x=0 TO 4
      LCDCMD LCDPin ,BorrarLCD                     'Limpiar pantalla
      LCDOUT LCDPin ,MoverCrsr,["PANTALLA LCD"]    'Enviar el mensaje a la línea 0
      PAUSE 300
      LCDCMD LCDPin ,BorrarLCD                     'Limpiar pantalla
      LCDOUT LCDPin ,MoverCrsr+64,["PANTALLA LCD"] 'Enviar el mensaje a la línea 1
      PAUSE 300
    NEXT
    PAUSE 300
    FOR x=0 TO 5                                   'Genera el efecto de parpadeo
    LCDCMD LCDPin ,LCDOFF                          'del mensaje en pantalla
    PAUSE 200
    LCDCMD LCDPin ,LCDON
    PAUSE 200
    NEXT
    GOTO Menu
```

Continuación del Programa 7_1.bsp

```
'================================================================================
'SUBRUTINAS

'Rutina de inicialización de la pantalla LCD en modo de 4 bits, 2 líneas
IniLcd:
  PAUSE 1000
  LCDCMD LCDPin ,Despertar :  PAUSE 10     'Despertar la pantalla
  LCDCMD LCDPin ,Despertar :  PAUSE 1      'Despertar la pantalla
  LCDCMD LCDPin ,Despertar :  PAUSE 1      'Despertar la pantalla
  LCDCMD LCDPin ,Modo4bits                 'Configurar el bus de dato en 4 bits
  LCDCMD LCDPin ,Modo2linea                'Se configura a dos líneas de 5 x 8
  LCDCMD LCDPin ,LCDOFF                    'Apagar la pantalla
  LCDCMD LCDPin ,LCDON                     'Encender la pantalla sin cursor
  LCDCMD LCDPin ,IncrCrsr                  'Ajusta la pantalla para auto-incremento
  LCDCMD LCDPin ,BorrarLCD                 'Limpia la pantalla
RETURN
```

El programa 7_1.bsp, muestra como efectuar las operaciones más simples con una pantalla LCD, es decir, inicializarla y mostrar en ella una serie de caracteres. Merece especial atención la rutina de inicialización, pues de su correcto funcionamiento dependerá que la pantalla reconozca los comandos y datos que le serán enviados posteriormente. Si no puede lograr una visualización correcta de la pantalla, ajuste el contraste mediante el potenciómetro RP.

La figura No. 7.5. muestra el diagrama de flujo para inicializar una pantalla LCD en modo de 4 bits. La rutina IniLcd, hace uso de este diagrama para inicializar al LCD en modo de 4 bits, 2 líneas, fuente 5x8, pantalla sin cursor y con auto-incremento. IniLcd se inicia con una espera de 1000 ms, tiempo mucho mayor que el requerido, que garantiza la estabilización de la pantalla y de sus circuitos internos.

El programa presenta un menú al usuario y ejecuta una cierta acción de acuerdo a la selección que se efectúe. Si se selecciona la opción 1 del menú (BZ) se producirá un sonido de unos 4.5 Khertz. Al seleccionar la opción 2 (LED) se cambiará el estado del pin P12 del BASIC Stamp, encendiendo y apagando alternadamente al LED conectado a éste. La opción 3 desencadena un efecto mucho más llamativo al mostrar el mensaje **PANTALLA LCD** en las dos líneas del LCD para luego hacerlo parpadear.

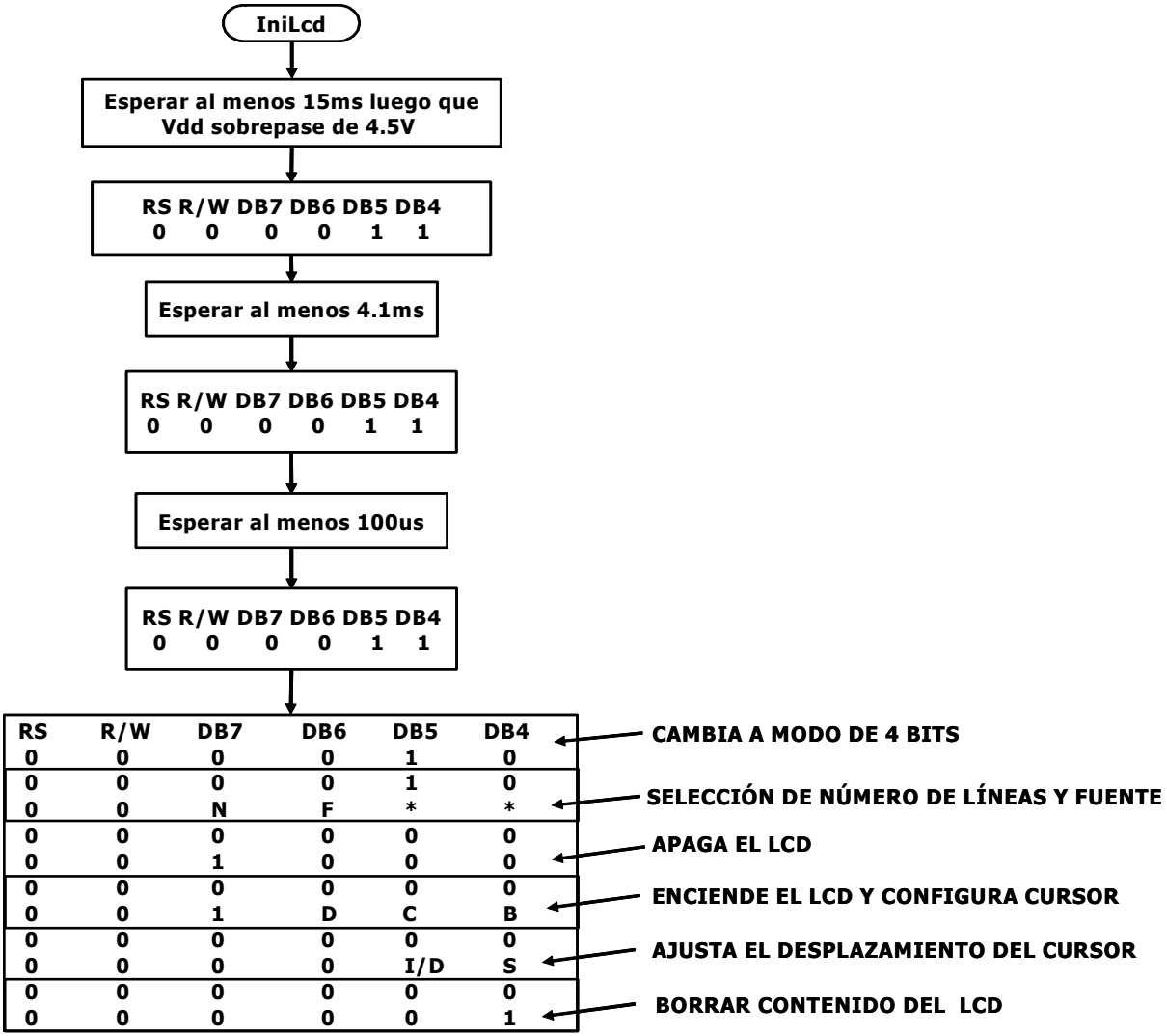

IniLcd
Esperar al menos 15ms luego que Vdd sobrepase de 4.5V
RS R/W DB7 DB6 DB5 DB4
0 0 0 0 1 1
Esperar al menos 4.1ms
RS R/W DB7 DB6 DB5 DB4
0 0 0 0 1 1
Esperar al menos 100us
RS R/W DB7 DB6 DB5 DB4
0 0 0 0 1 1
RS R/W DB7 DB6 DB5 DB4
0 0 0 0 1 0
CAMBIA A MODO DE 4 BITS
0 0 0 0 1 0
0 0 N F * *
SELECCIÓN DE NÚMERO DE LÍNEAS Y FUENTE
0 0 0 0 0 0
0 0 1 0 0 0
APAGA EL LCD
0 0 0 0 0 0
0 0 1 D C B
ENCIENDE EL LCD Y CONFIGURA CURSOR
0 0 0 0 0 0
0 0 0 0 I/D S
AJUSTA EL DESPLAZAMIENTO DEL CURSOR
0 0 0 0 0 0
0 0 0 0 0 1
BORRAR CONTENIDO DEL LCD

Figura No. 7.5: Diagrama de flujo para la inicialización de un LCD con el controlador HITACHI 44780.

Fuente: Fuente: Hitachi HD44780, Manual del usuario.

Ejercicio # 2: Acciones Básicas con el LCD

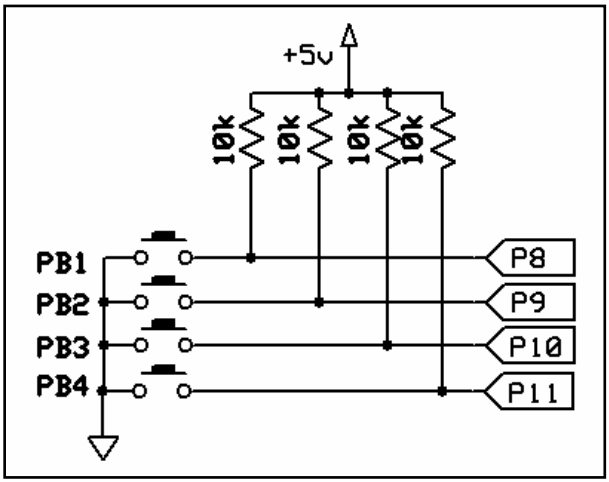
Objetivo: Utilizar el BASIC Stamp 2p para manipular una pantalla de cristal líquido paralela de 2x16 y configurarla en sus distintos modos de operación.

Elementos a Utilizar

(1) BASIC Stamp 2p.
(1) Un cable serial DB9.
(1) LCD alfanumérico 2x16 paralelo.
(1) Resistencia de 10 Kohms.
(1) Potenciómetro de 10 Kohms.
(1) Banco de Pulsadores.
(1) StampW (v2 Beta 2.1 o superior).

Monte el circuito de la figura No. 7.6 y proceda a descargar el programa 7_2.bsp.

En este programa se implementan las distintas instrucciones que soporta el controlador HITACHI 44780 que maneja al LCD. Entre las instrucciones se encuentran: desplazar el mensaje a la derecha, desplazar el mensaje a la izquierda, mover el cursor a la derecha, mover el cursor a la izquierda, posicionar el cursor en un carácter específico y las distintas configuraciones que se pueden aplicar al cursor de la pantalla. Todas estas acciones se consideran instrucciones básicas en el manejo de un LCD, por medio de ellas puede implementar efectos visuales muy útiles para su proyecto.



Cuando se trabaja con un LCD hay que optimizar el espacio visual proporcionado por la pantalla. Es necesario disponer del mismo de forma correcta a efectos de no confundir al usuario. También, es especialmente importante definir la cantidad y calidad de la información a desplegar.

Figura No. 7.6: Conexión de los pulsadores al Basic Stamp, para el ejercicio #2.
Fuente: Elaboración Propia.

```
'Programa 7_2.bsp
'Este programa tiene por objetivo mostrar como configurar en los distintos modos de
'operación de la pantalla LCD
'=================================================================================
'{$STAMP BS2p}
'{$PBASIC 2.5}
'
'Declaración de Variables y Constantes

BorrarLCD     CON %00000001       'Limpiar pantalla
MoverCrsr     CON %10000000       'Posicionar el cursor
Despertar     CON %00110000       'Despertar al LCD
Modo4bits     CON %00100000       'Modo de 4 bits de datos
Modo2linea    CON %00101000       'Modo de 2 líneas, tamaño de 5x8 (%00101100 5x10)
LCDOFF        CON %00001000       'Apagar la pantalla
LCDON         CON %00001100       'Encender, sin cursor la pantalla
LCDONCrsr     CON %00001110       'Encender, con cursor la pantalla
LCDONCrsrParp CON %00001111       'Encender, con cursor la pantalla
LCDONParp     CON %00001101       'Encender, sin cursor la pantalla
IncrCrsr      CON %00000110       'Auto-incremento del cursor
Moverpizq     CON %00011000       'Desplazar pantalla a la izquierda
Moverpder     CON %00011100       'Desplazar pantalla a la derecha
Movercizq     CON %00010000       'Desplazar cursor a la izquierda
Movercder     CON %00010100       'Desplazar cursor a la derecha
Linea2        CON %11000000       'Ir a la línea inferior de la pantalla
LCDpin        CON 0               'Terminal Enable de la LCD
Pulset1       VAR Byte            'Espacio de trabajo del pulsador 1
Pulset2       VAR Byte            'Espacio de trabajo del pulsador 2
Pulset3       VAR Byte            'Espacio de trabajo del pulsador 3
Pulset4       VAR Byte            'Espacio de trabajo del pulsador 4
S1            PIN 8               'Pulsador PB1
S2            PIN 9               'Pulsador PB2
S3            PIN 10              'Pulsador PB3
S4            PIN 11              'Pulsador PB4
x             VAR Byte            'Variable de uso general

'_________________________________________________________________________________
'Programa Principal:

Inicio:
  GOSUB IniLcd
```

Continuación del Programa 7_2.bsp

```
Menu:
  LCDCMD LCDpin,LCDON                          'Configuración Inicial
  LCDOUT LCDpin,MoverCrsr,["PROGRAMA 7_2.bsp"]
  LCDOUT LCDpin,Linea2,["PB1 = CONTINUAR"]
NPS1:
  BUTTON  S1,0,255,0,pulset1,0,NPS1            'Espera hasta que PB1 sea presionado
  PAUSE 100
  LCDCMD LCDpin,BorrarLCD                      'Limpiar la pantalla
  LCDOUT LCDpin,MoverCrsr,["1-SCur 2-CCur"]    'Permite desplegar el
  LCDOUT LCDpin,Linea2,["3-SCurB 4-CCurB"]     'siguiente menú de opciones

  DO
    BUTTON  S1,0,255,0,pulset1,1,SinCrsr       'Si PB1 fue presionado => Sin Cursor
    BUTTON  S2,0,255,0,pulset2,1,ConCrsr       'Si PB2 fue presionado => Con Cursor
    BUTTON  S3,0,255,0,pulset3,1,SinCrsrPar    'Si PB3 fue presionado => Con Parpadeo
    BUTTON  S4,0,255,0,pulset3,1,ConCrsrPar    'Si PB4 fue presionado => Sin Parpadeo
  LOOP

  SinCrsr:
    LCDCMD LCDpin,LCDOFF                       'Apagar pantalla
    LCDCMD LCDpin,LCDON                        'Encender la pantalla sin cursor
    GOSUB Efectos                              'Llamar a la subrutina Efectos
    GOTO Menu

  ConCrsr:
    LCDCMD LCDpin,LCDOFF                       'Apagar pantalla
    LCDCMD LCDpin,LCDONCrsr                    'Encender pantalla con cursor
    GOSUB Efectos                              'Llamar a la subrutina Efectos
    GOTO Menu

  SinCrsrPar:
    LCDCMD LCDpin,LCDOFF                       'Apagar pantalla
    LCDCMD LCDpin,LCDONParp                    'Encender pantalla con parpadeo activo,
    GOSUB Efectos                              'sin cursor
    GOTO Menu

  ConCrsrPar:
    LCDCMD LCDpin,LCDOFF                       'Apagar pantalla
    LCDCMD LCDpin,LCDONCrsrParp                'Encender pantalla con parpadeo activo,
    GOSUB Efectos                              'con cursor
    GOTO Menu
```

Continuación del Programa 7_2.bsp

```
'=================================================================================
'SUBRUTINAS
'La rutina Efectos se encarga de generar un conjunto de mensajes que demuestran la
'diferencia entre las opciones seleccionadas

Efectos:
  LCDCMD LCDpin,BorrarLCD                          'Limpiar la pantalla
  LCDOUT LCDpin,MoverCrsr,["1-Mover Pantalla"]     'Mostrar por pantalla las opciones
  LCDOUT LCDpin,Linea2,["2-Mover Cursor"]

  DO
    BUTTON  S1,0,255,0,pulset1,1,MoverPant         'PB1 presionado se desplaza pantalla
    BUTTON  S2,0,255,0,pulset2,1,MoverCr           'PB2 presionado se desplaza cursor
  LOOP

MoverPant:
  LCDCMD LCDpin,BorrarLCD                          'Limpiar pantalla
  LCDOUT LCDpin,MoverCrsr,["Desplazar LCD"]
  LCDOUT LCDpin,Linea2,["1-Izq 2-Der 3-Ex"]
  DO
    BUTTON  S1,0,250,100,pulset1,1,Despizq         'PB1 presionado, cursor a la der.
    BUTTON  S2,0,250,100,pulset2,1,Despder         'PB2 presionado, cursor a la izq.
    BUTTON  S3,0,255,0,pulset3,1,Salir             'Retornar al programa
  LOOP

Despder:
  LCDCMD LCDpin,BorrarLCD                          'Limpiar pantalla
  LCDOUT LCDpin,Linea2+3,["PRUEBA LCD.."]          'Mensaje de prueba
  PAUSE 500
  FOR x= 0 TO 14                                   'Ciclo para desplazamiento
    LCDCMD LCDpin,Moverpizq                        'Desplazar mensaje a la izquierda
    PAUSE 300
  NEXT
  FOR x= 0 TO 14                                   'Ciclo para desplazamiento
    LCDCMD LCDpin,Moverpder                        'Desplazar a la derecha el mensaje
    PAUSE 300
  NEXT
  GOTO MoverPant
```

Continuación del Programa 7_2.bsp

```
MoverCr:
  LCDCMD LCDpin,BorrarLCD                          'Limpiar la pantalla
  LCDOUT LCDpin,MoverCrsr,["1-Izq 2-Der 3-Ex"]  'Mostrar el siguiente menú de opciones
  LCDOUT LCDpin,Linea2+3,["PRUEBA LCD"]

Ciclo:
  BUTTON  S1,0,250,100,pulset1,1,Crizq          'PB1 presionado mover cursor a la der
  BUTTON  S2,0,250,100,pulset2,1,Crder          'PB2 presionado mover cursor a la izq
  BUTTON  S3,0,255,0,pulset3,1,Salir            'PB3 presionado retornar al programa
  GOTO Ciclo
Crder:
  LCDCMD LCDpin,Movercder                       'Desplazar cursor a la derecha
  PAUSE 300
  GOTO Ciclo

Crizq:
  LCDCMD LCDpin,Movercizq                       'Desplazar cursor a la izquierda
  PAUSE 300
  GOTO Ciclo

Salir:
  LCDCMD LCDpin,BorrarLCD                       'Limpiar la pantalla
  RETURN                                        'Retornar
'=================================================================================
'IniLcd inicializa la pantalla en modo 4 bits, 2 líneas

IniLcd:
  PAUSE 1000
  LCDCMD LCDpin,Despertar      'Despertar la pantalla
  PAUSE 10
  LCDCMD LCDpin,Despertar      'Despertar la pantalla
  PAUSE 1
  LCDCMD LCDpin,Despertar      'Despertar la pantalla
  PAUSE 1
  LCDCMD LCDpin,Modo4bits      'Configurar el bus de dato en modo de 4 bits
  LCDCMD LCDpin,Modo2linea     'Se configura a dos líneas de 5 x 8
  LCDCMD LCDpin,LCDOFF         'Apagar la pantalla
  LCDCMD LCDpin,LCDON          'Encender la pantalla sin cursor
  LCDCMD LCDpin,IncrCrsr       'Ajusta la pantalla para auto incremento
  LCDCMD LCDpin,BorrarLCD      'Limpia la pantalla
RETURN
```

Ejercicio # 3: Un Editor de Mensajes

Objetivos: Manipular una pantalla de cristal líquido paralela de 2x16, por medio de los comandos soportados por el microcontrolador BS2p para implementar un programa que permita editar un mensaje mostrado por la pantalla LCD. Utilizar los comandos LOOKUP y LOOKDOWN de PBASIC para buscar información en arreglos.

Elementos a Utilizar

(1) BASIC Stamp 2p.
(1) LCD alfanumérico 2 x 16 paralelo.
(1) Resistencia de 10 Kohms.
(1) Potenciómetro de 10 Kohms.
(1) Banco de Pulsadores.
(1) Un cable serial DB9.
(1) StampW (v2 Beta 2.1 o superior).

Mantenga en el protoboard del sistema de desarrollo MDK-B28 el esquema electrónico que se muestra en la figura No. 7.6. En el siguiente programa se implementará un sencillo editor de texto, donde, por medio de cuatro pulsadores, se le permitirá al usuario desplazarse dentro de la pantalla (en la primera fila) y modificar un texto de 16 caracteres. El pulsador 4 (PB4), ubicado en el terminal P11 permite mover a la derecha el cursor. El pulsador 3 (PB3), permite cambiar la letra en orden ascendente, es decir, A,B,C, etc. El pulsador 2 (PB2), permite cambiar la letra en orden descendente, es decir, C,B,A, ,Z….etc. El pulsador 1 (PB1), permite mover el cursor a la izquierda. Proceda a descargar en el BASIC Stamp el programa 7_3.bsp.

```
'Programa 7_3.bsp
'Este programa tiene por objetivo mostrar:
'1) El uso de los comandos LCDCMD y LCDOUT para enviar comandos y mensajes al LCD
'2) El uso del comando LCDIN para leer información de la memoria DDRAM
'2) El uso de los comandos LOOKUP Y LOOKDOWN
'=============================================================================
'{$STAMP BS2p}
'{$PBASIC 2.5}
'_____________________________________________________________________________
'Declaración de Variables y Constantes

BorrarLCD     CON %00000001       'Limpiar pantalla
MoverCrsr     CON %10000000       'Posicionar el cursor
Despertar     CON %00110000       'Despertar
Modo4bits     CON %00100000       'Modo de 4 bits de datos
Modo2linea    CON %00101000       'Modo de 2 líneas, tamaño de 5x8
LCDOFF        CON %00001000       'Apagar la pantalla
LCDON         CON %00001100       'Encender, con cursor la pantalla
IncrCrsr      CON %00000110       'Auto-incremento del cursor
InicioLCD     CON %00000010       'Posición inicial del LCD
LCDONCrsr     CON %00001110       'Encender, con cursor la pantalla
```

Continuación del Programa 7_3.bsp

```
EspBlanco     CON $10             'Espacio en blanco

Pulset1       VAR Byte            'Espacio de trabajo de PB1
Pulset2       VAR Byte            'Espacio de trabajo de PB2
Pulset3       VAR Byte            'Espacio de trabajo de PB3
Pulset4       VAR Byte            'Espacio de trabajo de PB4
Letra         VAR Byte            'Contiene el dato a mostrar
Pos           VAR Byte            'Guarda la posición en la pantalla
x             VAR Byte            'Variable de uso general

LCDpin        PIN 0               'Terminal Enable del LCD
PB1           PIN 8               'Pulsador PB1
PB2           PIN 9               'Pulsador PB2
PB3           PIN 10              'Pulsador PB3
PB4           PIN 11              'Pulsador PB4
'-------------------------------------------------------------------------------
'Programa Principal:

Inicio:
GOSUB IniLcd
GOSUB Mensaje_Inicial

DO
  BUTTON  PB4,0,255, 0 ,Pulset4,1,Derecha   'Si PB4 fue presionado va a la derecha
  BUTTON  PB3,0,254,254,Pulset3,1,Arriba    'Si PB3 fue presionado cambiar letra
  BUTTON  PB2,0,254,254,Pulset2,1,Abajo     'Si PB2 fue presionado cambiar letra
  BUTTON  PB1,0,255, 0 ,Pulset1,1,Izquierda 'Si PB1 fue presionado va a la izquierda
Salto:
LOOP

Derecha:
  IF Pos = 15 THEN                          'Si Pos es 15 implica el fin
    LCDOUT LCDpin,MoverCrsr+79,[EspBlanco]  'por la derecha, solo mostrar flecha
    LCDCMD LCDpin,MoverCrsr+Pos             'flecha a la izq.
  ELSE
    Pos = Pos+1                             'Incrementar posición a hacia la derecha
    GOSUB Mostrar_Flechas                   'Mostrar ambas flechas de desplazamiento
  ENDIF
  GOTO Salto
```

Continuación del Programa 7_3.bsp

```
Arriba:
  GOSUB Obtener_LCDChar                    'Lee del LCD la letra que se quiere editar
  x = (x+1) MAX 26                         'Determina cual es la nueva letra a mostrar
  GOSUB Mostrar_LCDChar                    'Presentar nueva letra en la LCD
  GOTO Salto

Abajo:
  GOSUB Obtener_LCDChar                    'Lee del LCD la letra que se quiere editar
  x = (x-1) MIN 0                          'Determina cual es la nueva letra a mostrar
  GOSUB Mostrar_LCDChar                    'Presentar nueva letra en la LCD
  GOTO Salto

Izquierda:
  IF Pos = 0 THEN                                'Si pos es 0, implica que el fin de
    LCDOUT LCDpin,MoverCrsr+64,[EspBlanco]        'la pantalla por la izquierda, en
    LCDCMD LCDpin,MoverCrsr                       'cuyo caso, sólo se muestra la flecha
  ELSE                                            'a la derecha
    Pos = Pos-1                                   'Mover una posición a la izquierda
    GOSUB Mostrar_Flechas                         'Mostrar las dos flechas
  ENDIF
  GOTO Salto

'================================================================================
'SUBRUTINAS
'Inicializacion de la pantalla:
IniLcd:
  PAUSE 1000
  LCDCMD LCDpin,Despertar                 'Despertar la pantalla
  PAUSE 10
  LCDCMD LCDpin,Despertar                 'Despertar la pantalla
  PAUSE 1
  LCDCMD LCDpin,Despertar                 'Despertar la pantalla
  PAUSE 1
  LCDCMD LCDpin,Modo4bits                 'Configurar el bus de dato en modo de 4 bits
  LCDCMD LCDpin,Modo2linea                'Se configura a dos líneas de 5x8
  LCDCMD LCDpin,LCDOFF                    'Apagar la pantalla
  LCDCMD LCDpin,LCDONCrsr                 'Encender la pantalla con parpadeo activo
  LCDCMD LCDpin,IncrCrsr                  'Ajusta la pantalla para auto-incremento
  LCDCMD LCDpin,BorrarLCD                 'Limpia la pantalla
RETURN
```

Continuación del Programa 7_3.bsp

```
'Mensaje a mostrarse al iniciar el programa
Mensaje_Inicial:
  LCDCMD LCDpin,BorrarLCD                          'Limpiar la pantalla
  LCDOUT LCDpin,MoverCrsr,["ROSO ELECTRIC S"]      'Envía el mensaje a la pos. indic.
  LCDOUT LCDpin,MoverCrsr+64,[$7F]                 'Enviar flecha a la izquierda
  LCDOUT LCDpin,MoverCrsr+79,[$7E]                 'Enviar flecha a la derecha
  LCDCMD LCDpin,InicioLCD                          'Ir al inicio de la pantalla (Home)
RETURN

'Obtiene el carácter visualizado en la posición actual del LCD
Obtener_LCDChar:
  x = 255
  LCDIN LCDPin, 0, [x]                             'Lee la letra en la pos actual LCD
  LOOKDOWN x,["ABCDEFGHIJKLMNOPQRSTUVWXYZ "],x    'Busca el índice de la letra
RETURN

'Muestra el carácter apuntado por x en la posición pos
Mostrar_LCDChar:
  LOOKUP x,["ABCDEFGHIJKLMNOPQRSTUVWXYZ "],Letra 'Muestra el carácter  x de la lista
  LCDOUT LCDpin,MoverCrsr+Pos,[Letra]              'Mostrar la letra
  LCDCMD LCDpin,MoverCrsr+Pos
RETURN

'Muestra las flechas que indican en cual sentido se puede seguir avanzando

Mostrar_Flechas:
  LCDOUT LCDpin,MoverCrsr+64,[$7F]
  LCDOUT LCDpin,MoverCrsr+79,[$7E]
  LCDCMD LCDpin,MoverCrsr+Pos
RETURN
```

EJERCICIO # 4: CARACTERES PERSONALIZADOS

Objetivo: Utilizar el Software "LCD Character Creador" para generar caracteres personalizados y visualizarlos en la pantalla. Construir una pequeña animación en el LCD.

Elementos a Utilizar

(1) BASIC Stamp 2p.
(1) Módulo LCD alfanumérico 2x16 paralelo.
(1) Software LCD Character Creator.
(1) Resistencia de 10 Kohms.
(1) Potenciómetro de 10 Kohms.
(1) StampW (v2 Beta 2.1 o superior).
(1) Cable serial DB9.

Monte en el protoboard que acompaña al sistema de desarrollo MDK-B28 el esquema de la figura No. 7.6.

Conecte el altavoz piezoeléctrico del sistema al pin P12 del BASIC Stamp.

LCD Character Creator

LCD Character Creator, es un programa propiedad de Parallax, Inc, el cual, facilita la creación de caracteres personalizados en pantallas LCD que utilicen el controlador HD44780 de HITACHI o algún otro compatible con éste.

En los ejercicios anteriores se ha manejado la pantalla LCD de casi todas las formas posibles, pero se ha hecho uso exclusivo del conjunto de caracteres almacenado en la memoria CGROM propia del controlador de la pantalla. Ahora bien, en este ejercicio vamos a crear nuestro propio conjunto de caracteres, para de esta forma ,mejorar del despliegue visual en la pantalla LCD.

La pantalla LCD dispone de un espacio de memoria RAM, en el cual, se pueden almacenar hasta 8 caracteres personalizados de 5x8 píxeles. Este espacio es llamado CGRAM. El contenido de la CGRAM puede ser modificado dinámicamente durante la ejecución del programa, razón por la cual, es ilimitado el número de caracteres que se pueden crear y mostrar, considerando claro que cada uno de ellos ocupa 8 bytes (en caso de ser fuente 5x8, si se usa fuente 5x11 serán 11 bytes por carácter) de la memoria del microcontrolador.

La creación de un carácter es sencilla si utilizamos el programa "LCD Character Creador". Tomando como referencia la figura No. 7.7, podemos darnos cuenta que cada fila de la matriz representa un byte donde cada columna es un bit del mismo. El valor lógico de cada bit se relaciona con el estado de cada píxel, de modo que si el correspondiente bit es uno, el píxel se mostrará oscuro (opaco), de lo contrario será claro (transparente).

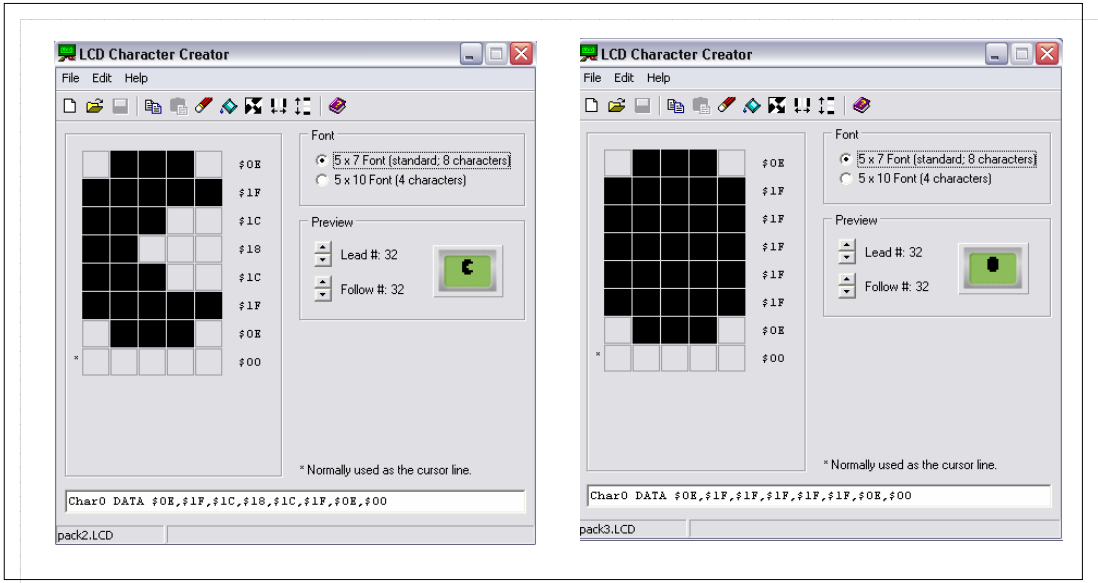


Figura No. 7.7: Ejemplos de creación de caracteres con el programa LCD Character Creator.
Fuente: Elaboración Propia.

Introduzca el programa 7_4.bsp, ejecútelo y analice su funcionamiento.

```
'Programa 7_4.bsp
'Este programa tiene por objetivo:
'1) Utilizar la directiva DATA para almacenar información en la EEPROM del BS2p
'2) Visualizar en la pantalla LCD caracteres personalizados
'3) Crear una animación sencilla

'=========================================================================================
'{$STAMP BS2p}
'{$PBASIC 2.5}
'-----------------------------------------------------------------------------------------
```

Continuación del Programa 7_4.bsp

```
'Declaración de Variables y Constantes

Pack0 DATA $0E,$1F,$1C,$18,$1C,$1F,$0E,$00 'Carácter # 1
Pack1 DATA $0E,$1F,$1F,$18,$1F,$1F,$0E,$00 'Carácter # 2
Pack2 DATA $0E,$1F,$1F,$1F,$1F,$1F,$0E,$00 'Carácter # 3
Pack3 DATA $0E,$1F,$1F,$03,$1F,$1F,$0E,$00 'Carácter # 4
Pack4 DATA $0E,$1F,$07,$03,$07,$1F,$0E,$00 'Carácter # 5

BorrarLCD     CON %00000001       'Limpiar pantalla
MoverCrsr     CON %10000000       'Posicionar el cursor
Despertar     CON %00110000       'Despertar
Modo4bits     CON %00100000       'Modo de 4 bits de datos
Modo2linea    CON %00101000       'Modo de 2 líneas, tamaño de 5x8 (%00101100 5x10)
LCDOFF        CON %00001000       'Apagar la pantalla
LCDON         CON %00001100       'Encender, sin cursor la pantalla
IncrCrsr      CON %00000110       'Auto-incremento del cursor
CGRam         CON %01000000       'Apuntador a la CGRAM
Linea1        CON %10000000       'Línea 1 de la pantalla
Linea2        CON %11000000       'Línea 2 de la pantalla

Dir           VAR Byte            'Dirección de la EEPROM a leer
Caracter      VAR Byte            'Bytes del carácter
Verificar     VAR Byte            'Bytes escritos en CGRAM
Pos           VAR Byte            'Posición en la pantalla
Ani           VAR Byte            'Caracteres de la animación

LCDpin        CON  0              'Pin de habilitación del LCD
'_____________________________________________________________________________
'Programa Principal

Inicio:
    PAUSE 500
    GOSUB IniLcd                'Inicializa al LCD
    LCDCMD LCDpin,BorrarLCD      'Borrar pantalla
    LCDCMD LCDpin,CGRam          'Apuntar a CGRAM
    FOR Dir = Pack0 TO Pack4+7
      READ Dir,Caracter          'Leer caracteres desde EEPROM
      LCDOUT LCDpin,0,[Caracter] 'Cargar los 5 caracteres en CGRAM
    NEXT
```

Continuación del Programa 7_4.bsp

```
DO
   LCDCMD LCDpin,MoverCrsr: LCDOUT LCDpin,Linea1+3,["CARACTERES"]
   LCDOUT LCDpin,Linea2+1,["PERSONALIZADOS"]
   'Mensaje inicial a ser devorado
    FOR Pos=0 TO 15                                  'Desplazamiento por toda la línea
      FOR Ani=0 TO 2                                 'Ciclo para producir
        LCDOUT LCDpin,Linea1+Pos,[Ani]: PAUSE 100    'animación en sentido
      NEXT                                           'izquierda a derecha
      LCDOUT LCDpin,Linea1+Pos,[" "]                 'Borrado de posición
    NEXT
    LCDCMD LCDpin,MoverCrsr
    FOR Pos=15 TO 0
      FOR Ani=2 TO 4                                 'Ciclo para producir
        LCDOUT LCDpin,Linea2+Pos,[Ani]: PAUSE 100    'animación en sentido
      NEXT                                           'derecha a izquierda
      LCDOUT LCDpin,Linea2+Pos,[" "]
    NEXT
LOOP

'_____________________________________________________________________________
'SUB-RUTINAS
'Inicializacion de la pantalla:
 IniLcd:
  PAUSE 500
  LCDCMD LCDpin,Despertar :  PAUSE 10  'Despertar la pantalla
  LCDCMD LCDpin,Despertar :  PAUSE 1   'Despertar la pantalla
  LCDCMD LCDpin,Despertar :  PAUSE 1   'Despertar la pantalla
  LCDCMD LCDpin,Modo4bits              'Configurar el bus de dato en modo de 4 bits
  LCDCMD LCDpin,Modo2linea             'Se configura a dos líneas de 5x8
  LCDCMD LCDpin,LCDOFF                 'Apagar la pantalla
  LCDCMD LCDpin,LCDON                  'Encender la pantalla sin cursor
  LCDCMD LCDpin,IncrCrsr               'Ajusta la pantalla para auto-incremento
  LCDCMD LCDpin,BorrarLCD              'Limpia la pantalla
RETURN
```

En la pantalla LCD debe aparecer el mensaje (CARACTERES PERSONALIZADOS) que será literalmente devorado por el conocido personaje de video juego (Packman). Para lograr la animación se utilizan los caracteres de la figura No. 7.7. (sentido derecha a izquierda), mostrándolos uno a continuación del otro en un intervalo de 100 ms para producir la sensación de movimiento.

Figura 7.8:
Visualización del programa 7_4.bsp visto en una pantalla LCD de 16 caracteres x 4 líneas.
Fuente: Elaboración Propia.

Para la mayor comprensión del proceso de carga y utilización de los caracteres personalizados, se incluye el diagrama de flujo mostrado en la figura No. 7.9.

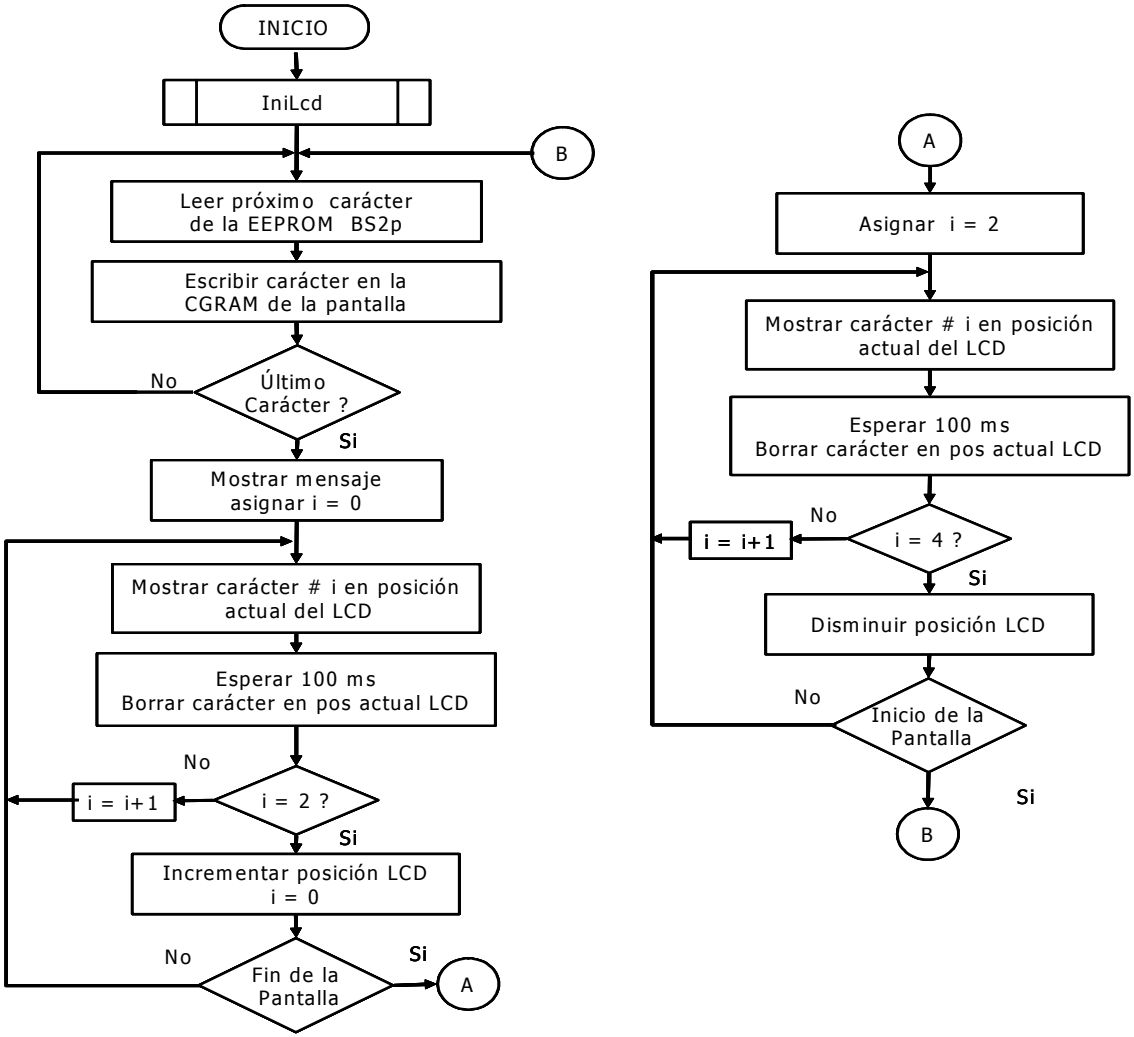


Figura No. 7.9: Diagrama de flujo para el programa 7_4.bsp.
Fuente: Elaboración Propia.

Analizando el programa 7_4.bsp, podemos entender como trabaja la memoria CGRAM y como puede almacenar los caracteres creados con el programa LCD Character Creator. Al mismo tiempo, se pudo implementar una animación con los caracteres personalizados. Si deseamos utilizar la pantalla LCD con otra animación, podemos introducir en el BASIC Stamp el programa 7_5.bsp.

EL programa 7_5.bsp, simula una medición de temperatura con indicación gráfica que activará una alarma visual y sonora cuando la variable temperatura llegue a valores predeterminados. Este programa deberá ser comprendido y asimilado completamente pues será la base para un ejercicio posterior.

Por sus dimensiones pudiera parecer que es un programa sumamente complejo, pero al comenzar el análisis, resultará evidente que las primeras 63 líneas del programa corresponden a la descarga de las pantallas en la memoria EEPROM del BASIC Stamp. El cuerpo principal del programa inicializa al LCD y escala a la variable x, de modo que su recorrido desde 0 hasta 255 sea interpretado como el desplazamiento a lo largo de las 15 pantallas que componen la animación. Posteriormente, se efectúan las llamadas a las subrutinas que muestran el valor de una temperatura. La figura No. 7.10 muestra el resultado de la ejecución de éste programa, utilizando el sistema de desarrollo MDK-B28 y una pantalla LCD.

Figura No. 7.10: Simulación en la LCD. Utilizando el programa 7_5.bsp.
Fuente: Elaboración Propia.

```
'Programa 7_5.bsp
'Este programa tiene por objetivos:
'1) La grabación y posterior lectura de datos en la EEPROM (DATA y READ)
'2) Descarga y visualización de caracteres personalizados
'3) Construcción de una animación compleja
'4) Actualizar la memoria CGRAM de una LCD dinámicamente
'================================================================================
'{$STAMP BS2p}
'{$PBASIC 2.5}
'_______________________________________________________________________________
'Las pantallas a mostrar son cargadas en la parte superior de la memoria EEPROM
'Pantalla #0
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$04,$08,$08,$08,$04,$03 : DATA $08,$08,$04,$02,$02,$02,$04,$18
'Pantalla #1
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$04,$08,$08,$08,$07,$03 : DATA $08,$08,$04,$02,$02,$02,$1C,$18
'Pantalla #2
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$04,$08,$08,$0F,$07,$03 : DATA $08,$08,$04,$02,$02,$1E,$1C,$18
'Pantalla #3
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$04,$08,$0F,$0F,$07,$03 : DATA $08,$08,$04,$02,$1E,$1E,$1C,$18
'Pantalla #4
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$04,$0F,$0F,$0F,$07,$03 : DATA $08,$08,$04,$1E,$1E,$1E,$1C,$18
'Pantalla #5
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$02,$07,$0F,$0F,$0F,$07,$03 : DATA $08,$08,$1C,$1E,$1E,$1E,$1C,$18
'Pantalla #6
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $02,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $08,$18,$1C,$1E,$1E,$1E,$1C,$18
'Pantalla #7
 DATA $03,$02,$02,$02,$02,$02,$02,$02 : DATA $18,$08,$08,$08,$08,$08,$08,$08
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
'Pantalla #8
 DATA $03,$02,$02,$02,$02,$02,$02,$03 : DATA $18,$08,$08,$08,$08,$08,$08,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
```

Continuación del Programa 7_5.bsp

```
'Pantalla #9
 DATA $03,$02,$02,$02,$02,$02,$03,$03 : DATA $18,$08,$08,$08,$08,$08,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
'Pantalla #10
 DATA $03,$02,$02,$02,$02,$03,$03,$03 : DATA $18,$08,$08,$08,$08,$18,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
 'Pantalla #11
 DATA $03,$02,$02,$02,$03,$03,$03,$03 : DATA $18,$08,$08,$08,$18,$18,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
 'Pantalla #12
 DATA $03,$02,$02,$03,$03,$03,$03,$03 : DATA $18,$08,$08,$18,$18,$18,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
 'Pantalla #13
 DATA $03,$02,$03,$03,$03,$03,$03,$03 : DATA $18,$08,$18,$18,$18,$18,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
 'Pantalla #14
 DATA $03,$03,$03,$03,$03,$03,$03,$03 : DATA $18,$18,$18,$18,$18,$18,$18,$18
 DATA $03,$03,$07,$0F,$0F,$0F,$07,$03 : DATA $18,$18,$1C,$1E,$1E,$1E,$1C,$18
'Pantalla de Alarma
UltimaP DATA $1C,$1D,$1D,$1D,$1D,$1D,$1D,$1D
        DATA $07,$17,$17,$17,$17,$17,$17,$17
        DATA $1D,$1D,$1B,$17,$17,$17,$1B,$1C
UltimoC DATA $17,$17,$1B,$1D,$1D,$1D,$1B,$07
'____________________________________________________________________________

'Declaración de Variables y Constantes
'Constantes de LDC

NoComando     CON %00000000
BorrarLCD     CON %00000001       'Limpiar pantalla
MoverCrsr     CON %10000000       'Posicionar el cursor
Despertar     CON %00110000       'Despertar
Modo4bits     CON %00100000       'Modo de 4 bits de datos
Modo2linea    CON %00101000       'Modo de 2 líneas, tamaño de 5x8 (%00101100 5x10)
LCDOFF        CON %00001000       'Apagar la pantalla
LCDON         CON %00001100       'Encender, sin cursor la pantalla
IncrCrsr      CON %00000110       'Auto-incremento del cursor
CGRam         CON %01000000       'Apuntador a la CGRAM
Linea1        CON %10000000       'Línea 1 de la pantalla
Linea2        CON %11000000       'Línea 2 de la pantalla
```

Continuación del Programa 7_5.bsp

```
'_____________________________________________________________________________

LCDpin        PIN  0              'Conexión del LCD
Alarmapin     PIN  12
Dir           VAR Word           'Dirección de la EEPROM a leer
Caracter      VAR Byte           'Bytes del carácter
Pos           VAR Byte           'Posición en la pantalla
x             VAR Byte           'Variable que simula la medición
i             VAR Nib            'Repetición de Señal de Alarma
Pantalla      VAR Nib            'Pantallas que componen la animación

'_____________________________________________________________________________
'Programa Principal
Inicio:
  PAUSE 500
  GOSUB InitLCD                  'Rutina de inicialización del LCD

DO
  FOR x=0 TO 255                 'Simulación de la medición de temperatura
    Pantalla=x/18                'Equivalencia entre la medición y el termómetro
    GOSUB Cargar_Pantalla        'Carga de la pantalla según temperatura
    GOSUB Actualizar_Pantalla    'Muestra la pantalla adecuada
    GOSUB Actualizar_Texto       'Actualización de mensajes
    PAUSE 250
  NEXT
LOOP
'=============================================================================
'SUBRUTINAS
'Inicializacion de la pantalla
InitLCD:
  PAUSE 1000
  LCDCMD LCDpin,Despertar :  PAUSE 10  'Despertar la pantalla
  LCDCMD LCDpin,Despertar :  PAUSE 1   'Despertar la pantalla
  LCDCMD LCDpin,Despertar :  PAUSE 1   'Despertar la pantalla
  LCDCMD LCDpin,Modo4bits              'Configurar el bus de dato en modo de 4 bits
  LCDCMD LCDpin,Modo2linea             'Se configura a dos líneas de 5x8
  LCDCMD LCDpin,LCDOFF                 'Apagar la pantalla
  LCDCMD LCDpin,LCDON                  'Encender la pantalla sin cursor
  LCDCMD LCDpin,IncrCrsr               'Ajusta la pantalla para auto-incremento
  LCDCMD LCDpin,BorrarLCD              'Limpia la pantalla
  RETURN
```

Continuación del Programa 7_5.bsp

```
'Rutina para cargar la pantalla en la GCRAM
Cargar_Pantalla:
  LCDCMD LCDpin,CGRam                  'Iniciar en carácter 0 de CGRAM
  FOR Dir = 32*Pantalla TO (32*(Pantalla+1)-1)
  'Lazo iterativo para leer desde la EEPROM del BS, los 4 caracteres de cada pantalla
   READ Dir,Caracter
   LCDOUT LCDpin,NoComando,[Caracter]
  NEXT
  RETURN

'Rutina para actualizar la animación, muestra los caracteres almacenados en la CGRAM
'en la pantalla

Actualizar_Pantalla:
  LCDOUT LCDpin,Linea1+1  ,[0] : LCDOUT LCDpin,Linea1+2 ,[1]  'Termómetro lado Izq
  LCDOUT LCDpin,Linea2+1  ,[2] : LCDOUT LCDpin,Linea2+2 ,[3]
  LCDOUT LCDpin,Linea1+13 ,[0] : LCDOUT LCDpin,Linea1+14,[1]  'Termómetro lado Der
  LCDOUT LCDpin,Linea2+13 ,[2] : LCDOUT LCDpin,Linea2+14,[3]
  RETURN

'Rutina para actualizar el texto de acuerdo a la medición de temperatura

Actualizar_Texto:
  LCDOUT LCDpin,Linea2+5,[DEC3 x,$20,$DF,"C"]
  SELECT Pantalla
    CASE 0 TO 2
      LCDOUT LCDpin,Linea1+4,["  Frio  "]
    CASE 3 TO 6
      LCDOUT LCDpin,Linea1+4,[" Tibio  "]
    CASE 7 TO 10
      LCDOUT LCDpin,Linea1+4,["Caliente"]
    CASE 11 TO 13
      LCDOUT LCDpin,Linea1+4,["Muy  Cal"]
    CASE ELSE
      GOSUB Alarma
  ENDSELECT
  RETURN
```

Continuación del Programa 7_4.bsp

```
'Rutina que genera una Señal de Alarma
Alarma:
  LCDCMD LCDpin,CGRam
    FOR Dir = 0 TO 31
      READ Dir,Caracter
      LCDOUT LCDpin,NoComando,[Caracter]
    NEXT
  LCDCMD LCDpin,CGRam+32
    FOR Dir = UltimaP TO (UltimoC+7)
      READ Dir,Caracter
      LCDOUT LCDpin,NoComando,[Caracter]
    NEXT
  FOR i=0 TO 8
    LCDOUT LCDpin,Linea1+1,[0]: LCDOUT LCDpin,Linea1+13,[0]:
    LCDOUT LCDpin,Linea1+2,[1]: LCDOUT LCDpin,Linea1+14,[1]
    LCDOUT LCDpin,Linea2+1,[2]: LCDOUT LCDpin,Linea2+13,[2]:
    LCDOUT LCDpin,Linea2+2,[3]: LCDOUT LCDpin,Linea2+14,[3]

    FREQOUT Alarmapin,943,1194
    LCDOUT LCDpin,128+4,[" Peligro "]
    LCDOUT LCDpin,Linea1+1,[4]: LCDOUT LCDpin,Linea1+13,[4]:
    LCDOUT LCDpin,Linea1+2,[5]: LCDOUT LCDpin,Linea1+14,[5]
    LCDOUT LCDpin,Linea2+1,[6]: LCDOUT LCDpin,Linea2+13,[6]:
    LCDOUT LCDpin,Linea2+2,[7]: LCDOUT LCDpin,Linea2+14,[7]
    FREQOUT Alarmapin,943,1061
  NEXT
RETURN
```

EVALUACIÓN DEL CAPÍTULO:

Investigue los siguientes tópicos

1. ¿ Qué es una pantalla LCD de caracteres y para que sirve ?.
2. ¿ Cuantos tipos de pantallas LCD conoce usted, explique ?.
3. ¿ Cual es la diferencia entre una pantalla de LCD de caracteres y una gráfica ?.
4. ¿ Cual es la diferencia entre una pantalla de LCD paralela y una serial ?.
5. ¿ Qué es la memoria CGRAM y cual es su utilidad ?.
6. ¿ Qué es la memoria CGROM y cual es su utilidad ?.
7. ¿ Cuales son las posiciones visibles y no visibles en una pantalla LCD ?.
8. ¿ Cual es la diferencia entre un comando y un dato para una pantalla LCD ?.
9. ¿ Por qué se tiene que inicializar a una pantalla LCD ?.
10. ¿ Qué comandos debo utilizar para inicializar una pantalla LCD ?.
11. ¿ Para que sirve el Comando LCDIN ?.
12. ¿ Para que sirve el comando LCDOUT ?.
13. ¿ Cuantos registros tiene la memoria EEPROM en un BASIC Stamp 2p ?.
14. ¿ Para que se utiliza el comando DATA ?.
15. ¿ Explique como funcionan los comandos READ y WRITE ?.
16. ¿ Cuantas pantallas LCD se pueden conectar en forma simultánea en un BASIC Stamp 2p ?.

Programación

1. Construya un programa junto a sus subrutinas para colocar en una pantalla LCD de caracteres, tipo paralela, sólo su nombre en la primera línea. Nota: No deberá utilizar ninguno de éstos comandos: LCDIN, LCDOUT. Si puede utilizar el comando LCDCMD. Ahora repita el programa sin utilizar los comandos LCDIN, LCDOUT ni el comando LCDCMD.